IN THE CLAIMS:

Q2

1. (Currently Amended) An MPEG decoder for improving decoding efficiency by scaling the decoding of an encoded digital video signal, comprising:

means for decoding a compressed video data stream including a plurality of macroblocks, said decoding means being operative to output quantized data from said decoded data stream;

means for performing an inverse quantization operation on said quantized data;

means for performing an inverse discrete cosine transform (IDCT) of the output from said means for performing an inverse quantization operation;

means for extracting header information from said quantized data from said decoding means and for performing a prediction operation according to predetermined criteria, <u>wherein said prediction operation defines a computation load from header information selected from the group consisting of: macroblock-type, motion vector magnitude, motion vector count, non-zero discrete cosine transformer coefficients and a coded block pattern (CBP)</u>;

means for generating a motion compensated reference value based on said quantized data from said decoding means; and

means for adding the output from said IDCT means and said motion generating means to produce motion compensated pictures.

2. (Original) The decoder of claim 1, further comprising means for storing the output of said adding means.

3. (Currently Amended)The decoder of claim 1, wherein said extracting means being operative to execute said prediction operation according to said extracted header information of the plurality <u>of</u> said macroblocks.

A2

4. (Original) The decoder of claim 1, wherein said prediction operation defines the computation load of said IDCT means and said motion generating means.

5. (Original) The decoder of claim 1, wherein the computation load of said MPEG decoder is selectively adjusted by scaling said IDCT means and said motion generating means based on said prediction operation.

6. (Cancelled).

7. (Original) The decoder of claim 1, wherein said prediction operation is implemented using a processing unit and software which controls the operation of said processing unit.

8. (Currently Amended) A programmable video decoding system, comprising:

a variable length decoder (VLD) configured to receive and decode a stream of block-based data packets, said VLD being operative to output quantized data from said decoded data packets;

a complexity estimator configured to extract the header information from said block-based data packets and further configured to execute a video complexity algorithm based on said extracted header information, wherein said extracted header information is used to determine an associated computation load that are combined to determine a total computation load;

an inverse quantizer coupled to receive the output of said variable length decoder to operatively inverse quantize the quantized data received from said variable length decoder;

an inverse discrete cosine transformer (IDCT) coupled to the output of said inverse quantizer for transforming the dequantizer data from frequency domain to spatial domain;

a motion compensator (MC) configured to receive motion vector data from said quantized data and to generate a reference signal; and

an adder for receiving said reference signal and said spatial domain data from said IDCT to form motion compensated pictures.

9. (Original) The system of claim 8, further comprising a buffer configured to store the output of said adder.

10. (Original) The system of claim 8, wherein said complexity estimator being operative to executed said video complexity algorithms according to the header information of said decoded block-based data packets, and wherein the computation load of said decoding system is regulated by scaling said IDCT and said MC based on said video complexity algorithms.

11. (Original) The system of claim 10, wherein the computation load of said IDCT and said MC are selectively adjusted according to said video complexity algorithm.

12. (Original) The system of claim 8, wherein said extracted header information includes a macroblock-type, a motion vector magnitude, a motion vector count, non-zero discrete cosine transformer (DCT) coefficients, and a coded block pattern (CBP) number from said decoded block-based data packets.

13. (Currently Amended) The system of claim [1] 8, wherein said video complexity algorithm is implemented using a processing unit and software, which controls the operation of said processing unit.

14. (Currently Amended) A method for improving decoding efficiency of an encoded [data] <u>digital</u> video signal employing an MPEG digital video decoder having a variable length code (VLC) decoder, an inverse quantizer (IQ), an inverse discrete cosine transformer (IDCT), a motion compensator (MC), and an estimator, said method comprising the steps of:

receiving a compressed video data stream at said VLC decoder and producing decoded data therefrom;

retrieving header information from said decoded data;

calculating a total computation load (C_{est}) for said IDCT and said MC based on the classification of said header information by said estimator;

dequantizing said decoded data using said inverse quantizer (IQ) to generate dequantized, decoded data;

employing said IDCT for transforming said dequantized, decoded data from frequency domain to spatial domain based [to] <u>on</u> said total computation load (C_{est}) to produce difference data;

employing said MC for generating a reference data of said encoded digital video signal based on said computation load; and

combining said reference data and said difference data to produce motion compensation pictures.

15. (Currently Amended) The method of claim 14, further comprising the step of:

transmitting said total computation load (C_{est}) to said IDCT and said MC.

16. (Original) The method of claim 14, wherein the step of calculating said total computation load (C_{est}) further comprising the steps of:

determining a first computation load (C_{type}) based on the macroblock of said retrieved header information;

a2

determining a second computation load (C_{mv}) based on the motion magnitude of said retrieved header information;

determining a third computation load (C_{mvc}) based on the motion vector magnitude of said retrieved header information;

determining a fourth computation load (C_{BN}) based on the number of non-zero DCT coefficients of said retrieved header information;

determining a fifth computation load (C_{CBP}) based on the coded block pattern (CBP) of said retrieved header information; and

combining said first, second, third, fourth, fifth computations loads, and an average computation load (C_{base}) to obtain said total computation load (C_{est}).

17. (Original) The method of claim 16, wherein the step of determining said first computation load (C_{type}) further comprising the steps of:

$C_{type} = 0$ if said retrieved header information provides a set of intra-coded coefficient;

$C_{type} = W$ if said retrieved header information provides a set of coded coefficient and non-motion compensated coefficient;

$C_{type} = 2W$ if said retrieved header information provides a set of motion compensated coefficient and non-coded coefficient; and

A2

$C_{type} = 3W$ if said retrieved header information provides a set of motion compensated coefficient and coded coefficient.

18. (Original) The method of claim 16, wherein the step of determining said second computation load (C_{mv}) further comprising the step of:

$C_{mv} = W_1$ if said retrieved header information provides a motion vector magnitude which is greater than a predetermined threshold otherwise $C_{mv} = 0$.

19. (Original) The method of claim 16, wherein the step of determining said third computation load (C_{mvc}) further comprising the step of:

$C_{mvc} = W_2$ x motion vector count (MV-Count) if said retrieved header information provides said motion vector count.

20. (Original) The method of claim 16, wherein the step of determining said fourth computation load (C_{BN}) further comprising the step of:

$C_{BN} = W_3$ x block number (BN) if said retrieved header information provides said block number.

21. (Currently Amended) The method of claim 16, wherein the step of determining said fifth computation load (C_{CBP}) further comprising the step of:

$C_{CBP} =$ [W4] $\underline{W_4}$ x number of non-zero coefficient if said retrieved header information provides said non-zero coefficient.

Q2

22. (Original) A prediction method of improving decoding efficiency of an encoded data video signal employing an MPEG digital video decoding system having a variable length code (VLC) decoder, an inverse quantizer (IQ), an inverse discrete cosine transformer (IDCT), a motion compensator (MC), and an estimator, said method comprising the steps of:

decoding a compressed bitstream including a plurality of macroblocks to obtain a corresponding decoded macro block;

obtaining a header classification criteria from the header information of said decoded macroblock;

employing said estimator for predicting a total computation load (C_{est}) according to the header information from said decoded macroblock and forwarding said total computation load (C_{est}) to said IDVCT and said MC; and

adjusting the computation load of said IDCT and said MC according to said total computation load (C_{est}).

23. (Original) The method of claim 22, further comprising the step of decoding said decoded macroblock according to said total computation load (C_{est}), together with motion vector information received in said compressed bitstream.

Q2

24. (Original) The method of claim 22, wherein said header classification criteria is defined as a macroblock-type, a motion vector magnitude, a motion vector count, non-zero discrete cosine transformer (DCT) coefficients, and a coded block pattern (CBP) number.

25. (Currently Amended) The method of claim 22, wherein the step of predicting said total computation load (C_{est}) further comprising the steps of:

determining a first computation load (C_{type}) based on the macroblock of the header information of said decoded macroblock;

determining a second computation load (C_{mv}) based on the motion magnitude of the header information of said decoded macroblock;

determining a third computation load (C_{mvc}) based on the motion vector magnitude of the header information of said decoded macroblock;

determining a fourth computation load (C_{BN}) based on the number of non-zero DCT coefficients of the header information of said decoded macroblock;

determining a fifth computation load (C_{CBP}) based on the coded block pattern (CBP) number of the header information of said decoded macroblock; and

combining said first, second, third, fourth, fifth computation loads, and an average computation load (C_{base}) to obtain said total computation load (C_{est}).

26. (Original) The method of claim 25, wherein the step of determining said first computation load (C_{type}) further comprising the steps of:

a2

$C_{type} = 0$ if the header information of said decoded macroblock provides a set of intra-coded coefficient;

$C_{type} = W$ if the header information of said decoded macroblock provides a set of coded coefficient and non-motion compensated coefficient;

$C_{type} = 2W$ if the header information of said decoded macroblock provides a set of motion compensated coefficient and non-coded coefficient; and

$C_{type} = 3W$ if the header information of said decoded macroblock provides a set of motion compensated coefficient and coded coefficient.

27. (Original) The method of claim 25, wherein the step of determining said second computation load (C_{mv}) further comprising the step of:

$C_{mv} = W_1$ if the header information of said decoded macroblock provides a motion vector magnitude which is greater than a predetermined threshold otherwise $C_{mv} = 0$.

28. (Original) The method of claim 25, wherein the step of determining said third computation load (C_{mvc}) further comprising the step of:

$C_{mvc} = W_2$ x motion vector count (MV-Count) if the header information of said decoded macroblock provides said motion vector count.

29. (Original) The method of claim 25, wherein the step of determining said fourth computation load (C_{BN}) further comprising the step of:

$C_{BN} = W_3$ x block number (BN) if the header information of said decoded macroblock provides said block number.

30. (Currently Amended) The method of claim 25, wherein the step of determining said fifth computation load (C_{CBP}) further comprising the step of:

C_{CBP} = [W4] $\underline{W_4}$ x number of non-zero coefficient if the header information of said decoded macroblock provides said non-zero coefficient.